

# Auction-Based Resource Allocation in Digital Ecosystems

Moreno Marzolla  
marzolla@cs.unibo.it

Stefano Ferretti  
sferrett@cs.unibo.it

Gabriele D'Angelo  
g.dangelo@unibo.it

Dipartimento di Scienze dell'Informazione  
Università di Bologna  
Mura A. Zamboni 7, I-40127 Bologna, Italy

## ABSTRACT

The proliferation of portable devices (PDAs, smartphones, digital multimedia players, and so forth) allows mobile users to carry around a pool of computing, storage and communication resources. Sharing these resources with other users (“Digital Organisms” – DOs) opens the door to novel interesting scenarios, where people trade resources to allow the execution, anytime and anywhere, of applications that require a mix of capabilities. In this paper we present a fully distributed approach for resource sharing among multiple devices owned by different mobile users. Our scheme enables DOs to trade computing/networking facilities through an auction-based mechanism, without the need of a central control. We use a set of numerical experiments to compare our approach with an optimal (centralized) allocation strategy that, given the set of resource demands and offers, maximizes the number of matches. Results confirm the effectiveness of our approach since it produces a fair allocation of resources with low computational cost, providing DOs with the means to form an altruistic digital ecosystem.

## Categories and Subject Descriptors

C.2.4 [Computer-Communication Networks]: Distributed Systems; H.m [Information Systems]: Miscellaneous

## Keywords

Resource Allocation, Optimization, Peer-to-Peer Systems, Ad-hoc Networks

## 1. INTRODUCTION

Mobile users are evolving: while in the recent past people used their mobile devices just for “simple” tasks such as checking e-mail or browsing the Web, the rise of novel social applications fosters a massive use of ubiquitous services. Current Operating Systems for mobile devices (e.g., Android, iOS) allow the execution of applications that, for instance, publish user’s geographical position and other context-aware information on social networking services. However, these forms of interaction are usually based on the classic client-server approach, i.e., the mobile device connects to a central service through its own Internet connection. The proliferation of heterogeneous devices with different capabilities (computation, communication, data storage, sensors and actuators) gives rise to new scenarios that promote the cooperation among individuals in order to guarantee the provision of “always on” services [6].

As an example, consider a medical doctor who receives an urgent call while attending a meeting with other colleagues. He needs to check some medical data to diagnose a particular illness; unfortunately, the tablet PC he carried with him has not enough memory and computational power to execute the job. Therefore, he “rents” CPU power from one of his colleagues high-end laptop to carry on the analysis.

As another example, consider a user that wants to upload some pictures made with her smart-phone/camera on Flickr or on the wall of her social Web application, but she is not provided with network connectivity. Hence she exploits the 3G mobile connection of a neighboring friend using that as a gateway to the net via an ad-hoc short range connection (e.g., Bluetooth).

In general, mobile users have many different devices in their pockets and suitcase, each of them with specific hardware and software characteristics. Quite often such devices are not enabled for seamless interaction with other devices belonging to the same owner; sharing resources among different people is even more challenging. The possibility for a user to exploit, in a Peer-to-Peer (P2P) and altruistic way, computing facilities owned by (known and trusted) neighbors requires mechanisms for automatic service discovery and negotiation, and for access control.

A system architecture supporting the scenarios above has been recently proposed [6]. Each mobile user is considered as a “digital organism” (DO), composed by many different devices belonging to the same human being. Each DO may share resources with peer DOs using auto-configuration strategies. Then, a community of interacting DOs can be thought of as a “digital ecosystem”. Each DO in the ecosystem contributes by providing its own unused resources to its (friend/trusted) neighbors. Therefore, the community of DOs exploits self-organization protocols and P2P strategies to create a cooperating, altruistic ecosystem.

Of course, privacy and security issues should be carefully considered, especially if data are to be distributed to other users. Addressing these issues in a mobile environment is highly nontrivial, especially when the remote nodes are completely untrusted. Therefore, we assume that each DO will preferably connect to other DOs towards which there is explicit trust (e.g., because users know each other). For example, a user traveling by train shall be willing to share a network connection with some (known and trusted) traveling companions.

In this paper we address the problem of optimizing the allocation of resources in a digital ecosystem, by matching resource requests and offers. We consider a P2P overlay which

connects DOs; each DO may offer and/or require resources, which are traded with neighboring DOs. We present a fully distributed scheme that can match demands and offers, allowing resources to be provisioned efficiently and fairly. We use a market-based approach in which requests are handled through ascending clock auctions [1]. We assume that each DO can use some form of “virtual currency” (tokens) as a form of payment for resources usage: this allows a fair allocation that balances supply and demand [15, 21]. First, we describe a distributed algorithm to carry on the auction; then, we formulate the resource allocation problem as a Mixed Integer Programming (MIP) optimization problem, which is used to compute the maximum number of requests that can be matched. We compare the optimal allocation with the one provided by our cheap, local strategy. Results show that our approach produces good allocations and requires low computational cost: this makes the auction-based allocation strategy particularly appropriate for sharing resources among devices with very limited computational power.

The rest of this paper is organized as follows. In Section 2 we give a precise formulation of the problem we are addressing. Section 3 presents our auction-based resource allocation scheme. In Section 4 we discuss numerical results obtained from a set of synthetic simulation experiments. In Section 5 we revise the literature and contrast our approach with some related work. Finally, concluding remarks are provided in Section 6. Additional details on the auction algorithm, and the MIP formulation of the optimization problem, are given in the Appendix.

## 2. PROBLEM FORMULATION

We consider a set  $\mathbf{R} = \{1, \dots, R\}$  of  $R$  different resource types (e.g., network connectivity, processing power, storage, and so on).  $\mathbf{N} = \{1, \dots, N\}$  denotes a set of  $N$  users trading these resources: each user can be either a *buyer* (if he requests resources) or a *seller* (if he offers resources). The same user may play the role of buyer and seller at the same time, offering surplus resources while buying those he needs.

For each user  $i \in \mathbf{N}$ , we denote with  $Req_{ri}$  the amount of type  $r$  resource requested by  $i$ ; for each  $j \in \mathbf{N}$ , we denote with  $Off_{rj}$  the amount of type  $r$  resource offered by  $j$ ; quantities are not restricted to be integer. The vectors  $Req_{\bullet i}$  and  $Off_{\bullet j}$  are called *resource bundles*<sup>1</sup>.

As an example, if there are two resource types (“CPU” and “Network Bandwidth”), then a resource bundle (0.1 MIPS, 200 KB/s) can be interpreted as a request (or an offer) for 0.1 MIPS of CPU power and 200 KB/s of network bandwidth. Unit of measures will be omitted in the following.

We assume that each user (node) is equipped with some form of wireless connectivity which enables short range interaction with a set of neighbors using an ad-hoc network topology. We model this with an  $N \times N$  adjacency matrix  $M_{ij}$ , where users  $i$  and  $j$  can interact iff  $M_{ij} = M_{ji} = 1$ ; matrix  $M_{ij}$  is symmetric, so that interactions are always bidirectional.

Each user can get resources from, or provide resources to, one of his direct neighbors; multi-hop interactions are not allowed. Multi-hop interactions would be much harder to

---

$\mathbf{N} :=$	$\{1, \dots, N\}$ set of users
$\mathbf{R} :=$	$\{1, \dots, R\}$ set of resource types
$Req_{ri} :=$	Amount of resource $r$ requested by user $i$
$Off_{rj} :=$	Amount of resource $r$ offered by user $j$
$RP_i :=$	Reserve price of buyer $i$ : maximum amount $i$ is willing to pay for his requested bundle
$SP_{rj} :=$	Unitary selling price of resource $r$ offered by user $j$
$M_{ij} :=$	1 iff $i$ can interact with $j$
$X_{irj} :=$	1 iff $i$ obtains resource type $r$ from $j$

---

**Table 1: Notation used in this paper**

handle, since appropriate routing strategies should be employed to ensure connectivity in spite of individual users moving and losing contact with neighbors. We introduce the binary decision variable  $X_{irj}$  which equals 1 iff user  $i$  obtains resource  $r$  from user  $j$ . If  $X_{irj} = 1$ , then  $i$  must obtain exactly  $Req_{ri}$  items of resource  $r$  from  $j$ . The allocation  $X_{irj}$  must satisfy the following constraints:

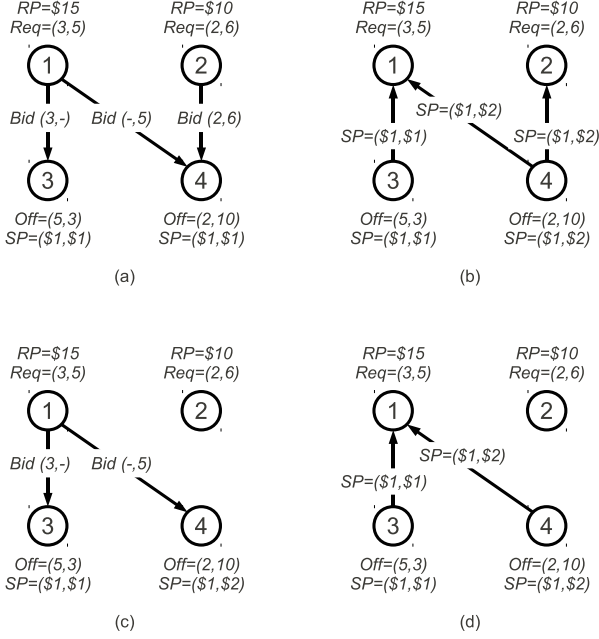
1. Each buyer  $i$  must obtain the requested quantities  $Req_{ri}$  of all resources  $r$  in his bundle, or none at all. Partially fulfilled requests are not allowed.
2. For each  $r$ , the requested quantity  $Req_{ri}$  must be provided by a single seller  $j$  (if the request is satisfied at all).
3. For each  $r$ , the offered quantity  $Off_{rj}$  can be fractioned across multiple buyers (i.e., a seller is not forced to provide all items of resource  $r$  to a single buyer).
4. If user  $i$  gets resource  $r$  from  $j$ , then the amount requested by  $i$  must not exceed the amount offered by  $j$ :  $Req_{ri} \leq Off_{rj}$ .
5. For all  $r \in \mathbf{R}$ ,  $\sum_{i \in \mathbf{N}} Req_{ri} X_{irj} \leq Off_{rj}$ , where the left-hand side represents the total amount of resource  $r$  provided by seller  $j$ . This means that the total amount of resource  $r$  provided by  $j$  to all buyers must not exceed his capacity.
6.  $X_{irj}$  can be 1 only if  $M_{ij} = 1$ : interactions are only allowed between neighbors.

The notation used in this paper is summarized in Table 1 (additional symbols shown in the table will be introduced in the next section).

The problem of finding an optimal allocation  $X_{irj}$  which maximizes the number of matched requests (i.e., maximizing the number of requests which can be satisfied by some seller) can be formulated as a MIP optimization problem (details are given in Appendix B). However, solving the optimization problem is impractical for several reasons: (i) global knowledge of all parameters is required, whereas each peer has only local knowledge; (ii) solving large instances of the optimization problem using general-purpose MIP solvers is time consuming; (iii) the optimal allocation may not even be desirable, since the constraints above do not take into account any measure of fairness between users. The lack of fairness is a particularly serious limitation, since it gives users no incentive to share their resources. While it would be possible to extend the optimization problem to take fairness into account, the other limitations would still apply.

<sup>1</sup>We use the symbol  $\bullet$  as a shortcut to denote a slice of a multi-dimensional vector; therefore,  $Req_{\bullet i}$  denotes the slice  $(Req_{1i}, Req_{2i}, \dots, Req_{Ri})$





**Figure 2: Example of resource allocation through ascending auction**

\$14, which is above the reserve price  $RP_2 = \$10$ . Therefore, user 2 gives up, while user 1 resubmit his bids (Figure 2(c)). Since there is no excess demands, the auction ends and user 1 can finally acquire his bundle (Figure 2(d)).

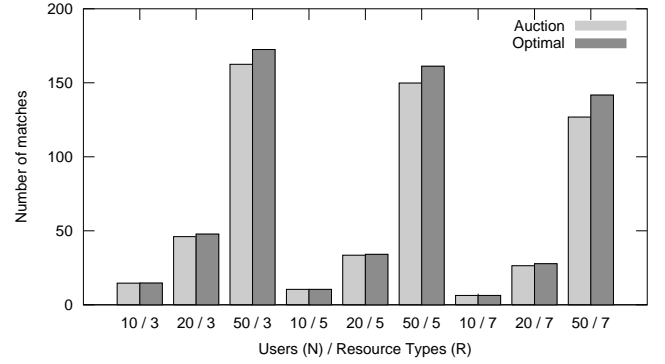
## 4. EXPERIMENTAL EVALUATION

In this section we analyze the performance of the auction-based resource allocation algorithm using a set of numerical experiments. We consider different network sizes (with  $N = 10, 20, 50$  users, respectively) and different numbers of resource types ( $R = 3, 5, 7$ ). For each combination of  $N$  and  $R$  we perform  $T = 10$  allocation steps. Each step involves the definition of requested and offered bundles (see below), and running an auction to match them. At the very beginning, each user is given a budget of 100 tokens; furthermore, before starting each step several initializations are performed, as follows.

First, we generate a random network with link density 0.3 (this means that on average, 30% of the elements of the adjacency matrix  $M_{ij}$  are nonzero). 20% of the users are randomly assigned the role of pure buyers (the offered bundles are set to zero), while the others are pure sellers (the requested bundles are set to zero). Each user is also assigned a random demand or offer vector: the number of items of each resource type that are offered/requested are drawn with uniform probability from the discrete set  $\{1, 2, \dots, 10\}$ . The initial reserve prices for sellers are set to 1, and the price increment is  $\Delta P = 1$ . The reserve price for buyer  $i$  is set to  $\epsilon_i \times \sum_{r \in \mathbf{R}} Req_{ri}$ , where  $\epsilon_i$  is uniformly chosen in  $[1.5, 2]$ ;  $\sum_{r \in \mathbf{R}} Req_{ri}$  is the cost of bundle  $Req_{\bullet i}$ , when all items have unitary cost. With the setup above we ensure that each node has sufficient liquidity to satisfy requests for resources, since each user will act most of the time as seller. In order to cope with statistical fluctuations, we executed 20 inde-

$N$	$R$	Matches		Auction/Optimal
		Auction	Optimal	
10	3	$14.60 \pm 1.48$	$14.70 \pm 1.45$	0.99
20	3	$46.10 \pm 2.33$	$47.80 \pm 2.41$	0.96
50	3	$162.50 \pm 4.55$	$172.50 \pm 3.14$	0.94
10	5	$10.40 \pm 1.07$	$10.40 \pm 1.07$	1.00
20	5	$33.50 \pm 2.93$	$34.10 \pm 3.12$	0.98
50	5	$149.80 \pm 4.44$	$161.20 \pm 2.58$	0.93
10	7	$6.30 \pm 1.64$	$6.30 \pm 1.64$	1.00
20	7	$26.40 \pm 3.30$	$27.80 \pm 3.50$	0.95
50	7	$126.80 \pm 6.41$	$141.70 \pm 5.38$	0.89

**Table 2: Number of matches (higher is better).**

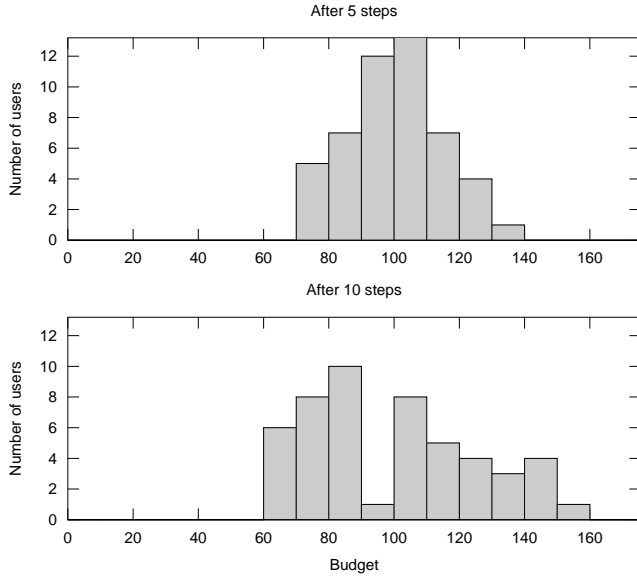


**Figure 3: Performance results: number of matches (higher is better)**

pendent replications of each sequence of  $T$  steps; at the end of all replications, average values and confidence intervals at  $(1 - \alpha) = 0.9$  confidence level are computed.

**Number of matches.** We first analyze the total number of matches, i.e., the total number of requests which can be satisfied at the end of the sequence of  $T$  allocation steps. We compare the value obtained using the auction with the optimal value obtained by matching requests using the MIP problem on Appendix B; the optimization problem has been solved using GLPK [16].

Raw results are reported in Table 2. The column labeled “Auction” shows the total number of matches at the end of the  $T$  auctions, computed with the ascending auction algorithm proposed in this paper. Column labeled “Optimal” shows the maximum number of matches computed using the optimization problem. The results has been plotted in Figure 3): we can see that the number of matches produced by the auction algorithm is only slightly less than the optimum value. It is important to report that for the larger systems ( $N = 50, R = 7$ ), GLPK required up to 10s to compute the optimal allocation (we used GLPK v4.45 on an AMD Athlon 64 X2 3800+ Dual Core Processor with 4 GB of RAM running Linux 2.6.32). The auction, implemented as a script in GNU Octave 3.2.3 [11], consistently requires less than a second on the same platform. We recall that the mechanism works on resource-constrained mobile devices composing DOs; hence, it is important to reduce as much as possible the overhead to perform the allocation. In



**Figure 4: Distribution of users budget, after 5 simulation steps (top) and at the end of the simulation (bottom);  $N = 50$ ,  $R = 7$**

Density	Matches		Price	Max Iter.
	Auction	Optimal		
0.20	103.40 $\pm$ 4.96	113.70 $\pm$ 4.14	1.07	6
0.40	141.00 $\pm$ 3.98	165.60 $\pm$ 3.27	1.15	18
0.60	132.30 $\pm$ 3.21	183.60 $\pm$ 1.82	1.28	29
0.80	89.60 $\pm$ 2.78	190.40 $\pm$ 1.57	1.48	56

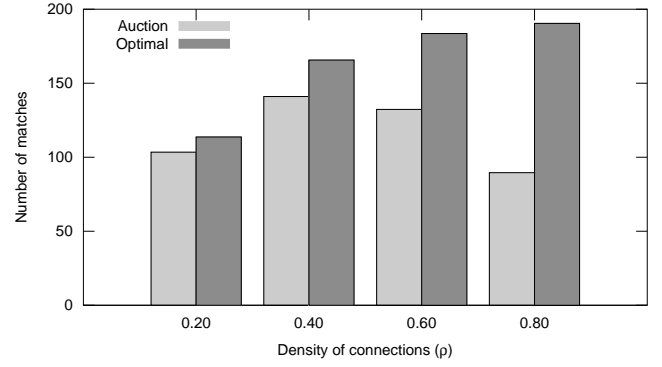
**Table 3: Number of matches for different connection densities;  $N = 50$ ,  $R = 7$**

practice we should expect some slowdown due to request-response messages sent through wireless links.

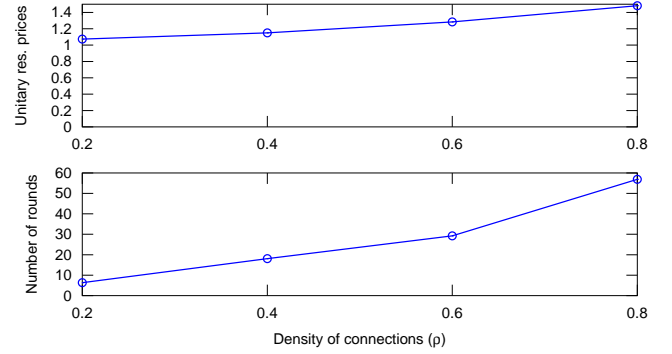
We show in Figure 4 the distribution of the users budget after 5 steps, and at the end of a simulation run (10 steps). Remember that each user is assigned an initial budget of 100 tokens. The budget distribution spreads over a larger interval as users trade resources, due to the nature of the experiments carried out. Each user has the same probability of being a buyer or a seller at each step as any other user.

**Behavior on crowded markets.** We also investigated the impact of the connection density (i.e., number of links between users) on the total number of matches. We consider  $N = 50$  users and  $R = 7$  resource types, and different values for the connection density  $\rho$  of the ad-hoc network. The value of  $\rho$  is the fraction of nonzero elements of the adjacency matrix  $M_{ij}$ . We considered  $\rho = 0.2, 0.4, 0.6, 0.8$ ; for each value, we executed  $T = 10$  allocation, and each sequence was independently repeated 20 times.

The raw data is shown in Table 3. As the connection density increases, each buyer can interact with more sellers, since each node has more neighbors. Therefore, we expect that the total number of matches increases because a buyer has more chances to find a seller with enough resources to match his demand.



**Figure 5: Number of matches for different connection densities;  $N = 50$ ,  $R = 7$**



**Figure 6: Average price for any single resource item (top) and average number of rounds to complete an auction (bottom) as a function of the connection density**

As we can see from Figure 5, the maximum number of matches indeed increases as  $\rho$  becomes larger. However, the total number of matches obtained from the auction has a maximum at about  $\rho \in [0.4, 0.6]$ , and starts decreasing afterwards. The bad behavior of the auction-based allocation can be explained by the fact that for large values of  $\rho$ , many buyers are likely to share the same neighboring sellers. Therefore, many buyers are likely also to share with other buyers the seller offering the lowest price: given that all buyers will bid the best (lowest) price, this will cause contention to the “best” seller.

To substantiate this claim, we show in Figure 6 the average price for a single resource item, and the average number of rounds which are necessary for the auction to settle to the final prices. We observe that both the average resource price and the number of rounds increases as the connection density  $\rho$  becomes larger. For  $\rho = 0.8$ , about 60 rounds are requested to conclude the auctions, resulting in higher unitary prices on average.

Several strategies can be used to mitigate the problem below: raising the prices by a quantity proportional to the excess demand or adding randomization in the choice of sellers (such that a buyer may occasionally bid to sub-optimal sellers) are possible extensions which are currently under investigation. However, we remark that large connection densities are somewhat unlikely in the scenario we are considering, as it would require a large number of people sharing

resources within a limited area. And furthermore, in such environments short-range communication technologies are preferred for their transmission power efficiency.

## 5. RELATED WORKS

The development of interaction mechanisms among wireless devices is a well studied area. For instance, the project “seamless computing” [4] proposed by Microsoft in 2003, and Jini, which is a part of the Java technology originally developed by Sun Microsystems, have addressed issues related to the definition and implementation of ubiquitous computing paradigms. Today, the notion of “digital ecosystem” is generally used to refer to any distributed system with properties of adaptability, self-organization, scalability and sustainability [3]. In this paper, we use this term in a slightly different acceptance, since we give more emphasis to issues concerned with mobile networks and interaction among constrained devices [6].

According to our view, there are two kinds of problems which need to be considered. The first one refers to the distributed resource utilization. In this sense, several works related to resource discovery, allocation and organization (mostly in ad-hoc fashion) are available in the literature, e.g. [5, 10, 13].

Another main problem is concerned with optimizing the communication capabilities of a DO. In general, this issue turns to allow a mobile node, having multiple wireless network interfaces, to change network points of attachment (handover) without disrupting existing connections, combined with the ability to disseminate messages in multi-hop transmissions (i.e., communication in a MANET). Examples of works on seamless host mobility are [2, 9, 17, 23].

As to the use of auctioning systems to allocate discrete computational resources, works have been already proposed, but usually employed on different use case scenarios, such as cluster and classic distributed systems [12, 8, 15, 19, 21]. Other works employ auction-based mechanisms in wireless networks; however, usually these are schemes that allow users to dynamically negotiate their agreed service levels with their service provider [7, 22], to define an optimal channel allocation or for scheduling. Hence, it is something very different from the P2P dynamic resource allocation we are considering in this work. For example, in [14], auction mechanisms are proposed to distributively coordinate and determine which nodes in a wireless network must act as relay nodes.

## 6. CONCLUSIONS

In this paper we presented a fully distributed algorithm for resource allocation between DOs. Our algorithm is based on ascending clock auctions, and allows users to trade resources in exchange for some form of digital currency. Numerical results show that this approach represents a viable and effective strategy promoting sharing of resources, thus providing DOs with the means to form an altruistic digital ecosystem.

As concerns the general deployment of the proposed scheme in a real distributed system, there are some open problems that require further investigation. Security issues are particularly important: for instance, authentication must be enforced in order to verify the identity of users that try to utilize resources of other DOs. We will also consider more general notations to describe requests for resources, e.g., re-

source bundles as intervals rather than single values.

## 7. REFERENCES

- [1] L. M. Ausubel and P. Cramton. Auctioning many divisible goods. *Journal of the European Economic Association*, 2(2-3):480–493, 2004.
- [2] M. Bonola, S. Salsano, and A. Polidoro. Upmt: universal per-application mobility management using tunnels. In *Proceedings of the 28th IEEE conference on Global telecommunications*, GLOBECOM’09, pages 2811–2818, Piscataway, NJ, USA, 2009. IEEE Press.
- [3] G. Briscoe. Complex adaptive digital ecosystems. In *Proc. International Conference on Management of Emergent Digital EcoSystems*, MEDES’10, pages 39–46, New York, NY, USA, 2010. ACM.
- [4] M. Chalmers and I. Maccoll. Seamless and seamless design in ubiquitous computing. In *Proceedings of Workshop At the Crossroads: The Interaction of HCI and Systems Issues in UbiComp. 2003*, 2003.
- [5] B. F. Cooper. An optimal overlay topology for routing peer-to-peer searches. In *Middleware ’05: Proceedings of the ACM/IFIP/USENIX 2005 International Conference on Middleware*, pages 82–101, New York, NY, USA, 2005. Springer-Verlag New York, Inc.
- [6] G. D’Angelo, S. Ferretti, V. Ghini, and F. Panzieri. Mobile computing in digital ecosystems: Design issues and challenges. volume *Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC 2011)*, July 2011.
- [7] M. Dramitinos, G. D. Stamoulis, and C. Courcoubetis. Auction-based resource allocation in umts high speed downlink packet access (hsdpa). In *in Proceedings of the First EuroNGI Conference on Traffic Engineering (NGI 2005)*, 2005.
- [8] D. F. Ferguson, C. Nikolaou, J. Sairamesh, and Y. Yemini. *Economic models for allocating resources in computer systems*, pages 156–183. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996.
- [9] S. Ferretti and V. Ghini. A web 2.0, location-based architecture for a seamless discovery of points of interests. In *Proceedings of the 2009 Fifth Advanced International Conference on Telecommunications*, pages 226–231, Washington, DC, USA, 2009. IEEE Computer Society.
- [10] G. H. L. Fletcher and H. A. Sheth. Unstructured peer-to-peer networks: Topological properties and search performance. In *3rd International Conference on Autonomous Agents and Multi-Agent Systems*, pages 14–27. Springer, 2004.
- [11] GNU Octave. <http://www.octave.org/>.
- [12] D. Grosu and A. Das. Auction-based resource allocation protocols in grids. In *In Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*, pages 20–27. ACTA Press, 2004.
- [13] I. Gruber, R. Schollmeier, and W. Kellerer. Peer-to-peer communication in mobile ad hoc networks. *Ad Hoc & Sensor Wireless Networks*, 2(3), 2006.
- [14] J. Huang, Z. Han, M. Chiang, and H. V. Poor. Auction-based distributed resource allocation for cooperation transmission in wireless networks. In





---

**Algorithm 2** SELLER( $j$ )

---

**Require:**  $Off_{\bullet j}$  Resource bundle offered by  $j$   
**Require:**  $SP_{\bullet j}$  Reserve prices of seller  $j$   
**Require:**  $\Delta P$  price increment  
1: Send  $SP_{\bullet j}, Off_{\bullet j}$  to all  $i \in \text{neighbors}(j)$   
2: **repeat**  
3:    $d_r := 0$  for all  $r \in \mathbf{R}$            {Demand for resource  $r$ }  
4:   **for all** bids  $Req_{ri}$  received from  $i$  **do**  
5:      $d_r := d_r + Req_{ri}$   
6:   **end for**  
7:    $exc := false$   
8:   **for all**  $r \in \mathbf{R}$  **do**  
9:     **if** ( $d_r > Off_{rj}$ ) **then**           {Excess demand?}  
10:        $SP_{rj} := SP_{rj} + \Delta P$   
11:        $exc := true$   
12:     **end if**  
13:   **end for**  
14:   Send  $SP_{\bullet j}, Off_{\bullet j}$  to all  $i \in \text{neighbors}(j)$   
15: **until** ( $exc = true$ )  
16: Receive  $Req_{ri} \times SP_{rj}$  tokens and allocate  $Req_{ri}$  items of resource  $r$  to buyer  $i$ ,  $r \in \mathbf{R}$

---

prices  $SP_{rj}$  (the minimum unitary price at which  $j$  is willing to sell resource  $r$ ), the offered quantity  $Req_{rj}$  and the price increment  $\Delta P$ . First, selling prices  $SP_{rj}$  are advertised to all neighbors (line 1). Then, the main loop starts; we use the vector  $d_r$  to keep track of the total demand of resource  $r$ , so that if excess demand for  $r$  is detected, its price  $SP_{rj}$  is incremented (line 10). The loop breaks only when there is no longer excess demand.

## B. OPTIMIZATION PROBLEM

In this section we formulate the problem of computing the maximum number of requests that can be satisfied as a MIP optimization problem. The optimization problem is the following:

Given:

$\mathbf{N} := \{1, \dots, N\}$  Set of users  
 $\mathbf{R} := \{1, \dots, R\}$  Set of resource types  
 $Req_{ri} :=$  Amount of resource  $r$  requested by  $i$   
 $Off_{rj} :=$  Amount of resource  $r$  offered by  $j$   
 $M_{ij} := 1$  iff user  $i$  can interact with user  $j$

Define:

$X_{irj} = 1$  iff user  $i$  gets resource  $r$  from user  $j$

Maximize:

$$\sum_{i \in \mathbf{N}} \sum_{r \in \mathbf{R}} \sum_{j \in \mathbf{N}} X_{irj} \quad (1)$$

Subject to:

$$\sum_{i \in \mathbf{N}} Req_{ri} X_{irj} \leq Off_{rj} \quad r \in \mathbf{R}, j \in \mathbf{N} \quad (2)$$

$$\sum_{j \in \mathbf{N}} X_{irj} \leq 1 \quad i \in \mathbf{N}, r \in \mathbf{R} \quad (3)$$

$$\sum_{j \in \mathbf{N}} X_{irj} = \sum_{j \in \mathbf{N}} X_{i1j} \quad i \in \mathbf{N}, r \in \mathbf{R} \quad (4)$$

$$X_{irj} \leq M_{ij} \quad i \in \mathbf{N}, r \in \mathbf{R}, j \in \mathbf{N} \quad (5)$$

The optimization problem above is a MIP problem since it involves binary decision variables  $X_{irj}$ .

Constraint (2) ensures that the total capacity of each seller  $j$  is not violated: the *total* amount of type  $r$  resource provided by  $j$  to all other users must not exceed its capacity  $Off_{rj}$ . Constraint (3) requires that each user  $i$  acquires resource type  $r$  from at most a single provider  $j$ ; note that  $i$  may be unable to acquire any resource at all, so the constraint is an inequality rather than an equality. Constraint (4) requires that, for each user  $i$ , either all the resources it needs are obtained, or none at all. Finally, constraint (5) requires that user  $i$  can request resources from user  $j$  ( $X_{irj} = 1$ ) only if  $i$  and  $j$  are neighbors ( $M_{ij} = 1$ ).

From constraint (4) we have that

$$\sum_{r \in \mathbf{R}} \sum_{j \in \mathbf{N}} X_{irj} = 0 \text{ or } R, \quad \text{for all } i \in \mathbf{N}$$

therefore, the total number of matches, i.e., the total number of buyers which can be satisfied, is

$$\sum_{i \in \mathbf{N}} \sum_{r \in \mathbf{R}} \sum_{j \in \mathbf{N}} \frac{X_{irj}}{R} \quad (6)$$

Since  $R$  is a constant, each assignment of  $X_{irj}$  maximizing (6) also maximizes the somewhat simpler expression (1) which we use as objective function.